A Proposal for Trust Management in Coalition Environments*

Dakshi Agrawal[†] Howard Chivers[‡] John Clark[♭] Charanjit Jutla[†]
John McDermid[♭]

[†] IBM T. J. Watson Research Center
P. O. Box 218
Yorktown Heights, NY 10598

[‡] Department of Computer Science
Cranfield University
Cranfield, UK

[♭] Department of Computer Science
University of York
York, UK

September 22, 2008

Abstract

It is well-recognized that for large catastrophes such as tornadoes, earthquakes, disease outbreaks, or aftermath of a war, coalitions that can coordinate and exploit resources and capabilities of many organizations are required to respond effectively. It is our thesis that in such coalition environments, classical security policies and access-control mechanisms need to be augmented by incorporating the notion of risk and trustworthiness of the parties involved. In this paper, we systematically analyze what trust is, highlight challenges in incorporating the notion of trust in coalition environments, and put forward a proposal to address these challenges.

1 Introduction

It is well-recognized that for large catastrophes such as tornadoes, earthquakes, disease outbreaks, or aftermath of a war, resources and capabilities of many organizations need to be coordinated to respond effectively [1]. In such situations, autonomous organizations form coalitions out of necessity and share their resources and capabilities with others to achieve common goals [3]. The member organizations in such coalitions can be of all sizes and capabilities—ranging from well-trained and well-equipped national forces to small ad hoc groups of individuals having unsurpassed knowledge of local geography, culture, needs, etc. Members join and depart coalitions based on individual goals, objectives, and capabilities [1].

Based on past experiences with the classical security policies and access-control mechanisms, it is clear that to support coalitions that can be rapidly created, modified, and dissolved, these mechanisms need to be augmented by incorporating the notion of risk and trustworthiness of the parties involved [7]. Traditionally, an examination of the centrally-issued credentials has been the usual method of certifying trustworthiness. However, such credentials are assigned after a time consuming and laborious process, and may not be available in a new theater of operation or when the span of operations stretches across continents. Even if the process of getting centrally-issued credentials was readily available, members of a coalition may be reluctant to go through the process for a variety of reasons (including privacy, cost, and perception of others). In such situations, a workable model of security that does not rely on centrally assigned credentials can enable a much higher degree of operational tempo.

There are two factors that contribute to the challenges in designing a workable trust management system for the kind of dynamic environment described above. First, when autonomous organizations have their own policies, risk postures, cultures that lead them to evaluate and perceive risk differently. Second, organizations may have relationship with one another outside the context of a coalition that needs to be taken into account. A trust management system must provide flexibility to accommodate these needs and be responsive to the dynamic coalition environments.

2 Contributions of This Paper

This paper provides a holistic view of trust in coalition environments by going back to the seminal works of McKnight and Chervany [9], and Hung, Dennis, and Robert [6]. We examine issues related to trust in the context of a coalition lifecycle. We then propose a new trust model and metric, inspired by the work of Reiter and Stubblebine [12], for trust management in coalition environments.

Our proposal is a departure from the traditional models of trust in a distributed network. The key difference is that our proposal does not require the path independence assumption to estimate risk inherent in a transaction as opposed to the traditional trust models [8]. We show that our proposal can be used to provide several other desirable properties in a trust management system including flexibility to accommodate heterogeneous coalition partners.

3 Trust

We start by providing a brief summary of the work by McKnight and Chervany [9], which contains a comparative study of sixty research articles or books, to provide an answer to the question "what is trust?" We follow this by a short summary of the work

*Research was sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defence and was accomplished under Agreement Number W911NF-06-3-0001. The views and conclusions contained in this document are those of the author(s) and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defence or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
01 DEC 2008	**N/A**	**-**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
A Proposal for Trust Management in Coalition Environments	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **IBM T. J. Watson Research Center P. O. Box 218 Yorktown Heights, NY 10598**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, distribution unlimited

13. SUPPLEMENTARY NOTES
See also ADM002187. Proceedings of the Army Science Conference (26th) Held in Orlando, Florida on 1-4 December 2008

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**UU**	**7**	

by Hung, Dennis, and Robert [6] to examine the issue of trust formation.

3.1 What is Trust?

It is generally agreed that trust is the expectation or likelihood of an agent behaving in a specific manner in a given context. Thus a *trust relationship* involves three entities: truster, trustee, and trust context. Depending on the trust context, a truster evaluates a trustee using subjective and objective criteria. In particular, based on their extensive survey, McKnight and Chervany [9] list four properties, benevolence, honesty, competence, and predictability that a truster evaluates before trusting an agent.

McKnight and Chervany describe a set of six inter-related *trust constructs* that are factors in forming a trust relationship between a truster and a trustee in a given context. The first factor is *situational trust* that describes the extent to which one intends to depend on a non-specific other party in a given situation. This could be a situation in which there is much to be gained from trusting but little attendant risk.

The second factor *system trust* describes the extent to which one agent believes that proper impersonal structures such as regulations, guarantees, contracts, or normality of situation, are in place to enable trustee to anticipate a successful future endeavor. Specifically, in the context of communication and networking systems, this will include trust placed in cryptosystems, intrusion-detection mechanisms, firewalls, etc.

The third factor *disposition trust* is a cross-situational, cross-personal construct that recognizes that agents will develop, over the course of their actions, generalized expectations about the trustworthiness of other agents. Thus, by definition, dispositional trust is a consistent tendency to trust across a broad spectrum of situations and persons. McKnight and Chervany provide an illustrative example, where when asked if he trusted his new manager, an employee said that he generally trusts new people, both at work, or elsewhere. We describe this situation as the aforementioned employee having a high degree of disposition trust.

Last three inter-related trust factors are *trusting belief, trusting intention, and trusting behavior*. *Trusting belief* means the extent to which one agent believes (and feels confident in believing) that the other person is benevolent, honest, competent, predictable, etc. in a given context[1].

Trusting intention means the extent to which one agent is *willing to depend* on another agent in a given context with a feeling of relative security, even though negative consequences are possible. Finally, *trusting behavior* means the extent to which one agent *voluntarily depends* on another agent in a given context with a feeling of relative security, even though negative consequences are possible[2].

Trusting behavior is the final result of interaction among all of the other trust constructs. Trusting behavior is often described quantitatively by *measurable indicators*, referred to in this paper as *trust metrics*. The main objective of a trust management system is to facilitate trusting behavior and provide trust metrics that can be used in risk assessments.

[1] In different contexts, different sets of properties, including those not listed above, may be important.

[2] The difference between trusting intention and trusting behavior is subtle—trusting intention supports trusting behavior, or in other words, trusting intention is an antecedent to trusting behavior. While this distinction is useful for and is a subject of psychological and sociological studies, for our purpose, it suffices to focus on trusting behavior.

3.2 How is Trust Formed?

Based on the dual process theories of cognition [11], Hung, Dennis, and Robert propose that trust behavior forms via three distinct routes at different stages of a trust relationship: the peripheral route, the central route, and the habitual route [6]. In the initial stages of a trust relationship, truster relies on peripheral cues (e.g., third party information, roles, categories); once there is a shared history between truster and trustee, they use the central route to evaluate properties such as honesty, competence, benevolence, and predictability to form trust; finally, after a long period of shared history, truster develops a habitual pattern of trust and is no longer interested in actively evaluating trust.

This discussion of trust brings out the importance of heterogeneity in coalition environments. Trusting behavior is *agent specific* and it depends on an agent's individual situation, disposition, beliefs, and intentions, and its past history and relationship with the trustee. While in e-commerce and peer-to-peer file sharing (eBay and Gnutella being the canonical examples, respectively), it is reasonable to assume that most benevolent agents are homogeneous, that is they largely have similar situation, disposition, beliefs, and intentions, the same is not true for coalition environments where even benevolent agents differ considerably. Thus, trust management systems from the e-commerce world cannot be deployed *talis qualis*.

4 Trust in Coalition Environments

In order to determine requirements for managing trust in a coalition environment, it is instructive to consider a model of collaboration lifecycle in a coalition mission and examine various aspects of collaboration relevant to forming trust. Clark *et al.* have proposed a model of the collaboration lifecycle which consists of the following seven phases [4]:

1. Identify mutual interest The first stage is the identification of the need as well as possibility of collaboration. In a dynamic situation, even in presence of need, a lack of prior trust relationship may inhibit collaboration. In such situations, a trust management system should provide a mechanism to establish initial trust.

2. Establish operational requirements and benefits Collaboration always involves an investment, if not in resources, then in time and opportunity cost, so it is usual to consider operational requirements and benefits of a collaboration. For the exposition in this paper, we will assume that the mission requires that coalition partners leverage each other's resources. Examples of these resources include different types of sensors and sensor platforms, personals with a certain set of skills, etc. We will assume that these resources cannot be substitutes for one another, e.g., an acoustic sensor cannot be substituted by a vibration sensor. We will also assume that it is possible to quantify these resources, e.g. five acoustic sensors, ten scouts, etc.

The next two phases are concerned with establishing the 'indirect cost' of collaboration.

3. Identify exposed assets and associated security impacts This is the first stage of a security assessment. It is used to establish the

assets (or classes of asset) that may be at risk, unwanted outcomes (e.g. unauthorized disclosure) and the impact of each unwanted outcome (e.g. prejudice operation, damage to the nation, cost). On indirect risk of a collaboration is dependency on resources provided by a coalition partner. In collaborative activities, a risk analysis necessarily needs to take into account the possibility that a coalition partner may not deliver the resources as agreed while the mission planning stage.

4. Establish possible threats and risk mitigation factors In a conventional risk assessment, the next stage is to identify threat paths and technical, management, or operational controls that can help mitigate the most significant risks.

Trust plays a crucial role at this stage. In particular, before a trusting behavior can be established, partners would assess various trust constructs as described in the previous section. Assessing these trust constructs would involve a range of factors including the likelihood of attackers within the collaborators user population, the quality of their system implementation, operation and management, agreement to process auditing, electronic system compliance monitoring, trusted computing etc. In a particular environment, the risk to security may turn out to be too high to be acceptable, and risk mitigation measures may need to be put in place before a collaboration can occur. Risk mitigation measures may include securing the possibility of using a backup in case a coalition partner fails to contribute as planned.

5. Agree the conduct of the collaboration A collaboration will usually develop a formal or informal operational agreement with an understanding that deviating from the agreed conduct may be penalized. Trust tokens can serve at this stage as a technical proxy for the agreed conduct of collaboration—trust token may provide assurance of a certain behavior and in turn, the system may guarantee that the conduct of collaborating parties would directly or indirectly impact content of their trust tokens.

6. Establish collaboration mechanisms The next stage is to make the collaboration operational. A large part of collaboration mechanisms are concerned with joint access control and usage of resources which includes authentication and identity management mechanisms.

7. Decommission the collaboration The final stage in the lifecycle is the decommissioning of a collaboration. From the trust management perspective, in this phase, history of collaboration needs to be stored, and feedback needs to be provided, for future evaluations of trust metrics.

5 A Proposal for Trust Management in Coalition Environments

The primary goal of a trust management system is to provide *trust metrics*, a measurable indicator of trust[3] among agents. As discussed in Section 3, trust is formed on the basis of a set of interrelated trust constructs. Among these trust constructs, situational trust, system trust, and disposition trust are *private properties* of the agent. We expect that partners in a coalition will have different degrees of disposition trust influenced by, and derived from, their respective institutionalized culture and policies formed as a result of their mission and past experiences. Similarly, situational trust and system trust will vary from agent to agent.

[3]According to McNight and Chervany trusting behavior is the measurable quantity. In this section, we will use the words trusting behavior and trust interchangeably.

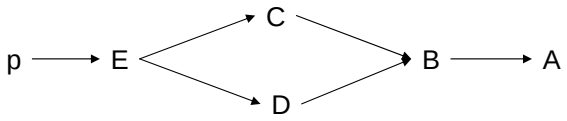

Figure 1: Recommendation and Consensus Operators

Among the remaining three trust constructs, trusting belief can be considered as the basic input to computing trust metrics[4]. Recall that trusting belief is the extent to which one agent believes that another agent is benevolent, honest, competent, predictable, etc. in a given context.

5.1 Traditional Trust Management Systems

In traditional trust management systems [13, 14], constructed typically for peer-to-peer systems or e-commerce, an agent A has two basic ways of establishing trusting belief in another agent P: past experiences of A in interactions with P or recommendation of P by other agents. This gives rise to *reputation graphs* which are used by traditional trust management system to compute trust metrics. We will discuss their suitability for coalition operations next.

To introduce reputation graphs, it convenient to use a simplified model of agent interactions. We model an agent P as a binary hypothesis; for example, the agent P provides either 'good' or 'bad' service. In Figure 1, the agent E has direct past experience with the agent P. Agents C and D have direct past experiences with how good is agent E's opinion of other agents. Similarly, the agent B has direct past experience with how good are agents C's and D's opinions of other agents.

Suppose the agent A wants to establish trusting belief in the agent P. Since the agent A does not have any direct past experience with P, it can send a query to the trust management system (or asks P itself), and discover that the agent E has direct past experience with P. However, if the agent E sends its estimate of P directly to A, then this estimate alone is useless for A since A does not have any experience in how good a recommendation from E is. Thus the problem of finding out quality of agent P reduces to the problem of finding out how good E's recommendation is. A can again send a query and discover that C and D have direct experience of how good E's recommendation is, and then the problem reduces to finding out how good C's and D's recommendations of other agents are. In this way, the trust evaluation continues recursively along all paths from P to A.

There are two basic operators over trust metrics that form the basis of the procedure given above, namely, *discounting* and a *consensus* operators [8]. Suppose A has a belief about B's capabilities, and further B has an opinion about C. From these two belief functions, one would like to derive a belief function which

[4]As commented earlier, we can ignore the subtle distinction between trusting intension and trusting behavior for the purpose of this paper.

reflects A's view about C (via B). This operation is called discounting and is denoted by $\otimes$; for example, in our case, $\omega_B^A \otimes \omega_C^B$, where ω represents the chosen trust metric[5], reflects A's view of C formed via B.

The second operator is required to derive the consensus of two possibly conflicting and uncertain opinions that reflects both opinions in a fair and equal way. This operator is called the consensus operator and is denoted by $\oplus$. For example, in Figure 1, B has two opinions, one from C and another from D about E; and in this case, $\omega_E^C \oplus \omega_E^D$, represents the trust metric obtained as a result of taking consensus of C and D's opinion of E.

Unfortunately, the discounting operator is not typically[6] distributive over the consensus operator. In Figure 1, B has an opinion about both C and D, and C and D have an opinion about E, and the latter in turn has an opinion about P. There are two possible ways to arrive at B's opinion about P. First, C and D could compute $\omega_P^C = \omega_E^C \otimes \omega_P^E$, and $\omega_P^D = \omega_E^D \otimes \omega_P^E$, respectively. The agent B can then receive opinons of C and D, and further discount it to compute $\omega_C^B \otimes \omega_E^C \otimes \omega_P^E$ and $\omega_D^B \otimes \omega_E^D \otimes \omega_P^E$. Finally, B can use consensus operator to combine these opinions, in effect computing

$$(\omega_C^B \otimes \omega_E^C \otimes \omega_P^E) \oplus (\omega_D^B \otimes \omega_E^D \otimes \omega_P^E) \tag{1}$$

Note that the calculation above can be carried out in a distributed fashion. However, the resulting trust metric is not accurate since, typically, the consensus operator $\oplus$ assumes that its operands are independent, while in (1), ω_P^E occurs in both operands of the consensus operator.

An alternative is to compute the more accurate expression

$$((\omega_C^B \otimes \omega_E^C) \oplus (\omega_D^B \otimes \omega_E^D)) \otimes \omega_P^E \tag{2}$$

which too is undesirable since it cannot be computed in a distributed manner, and as a result incurs computational and transmission overhead.

Most of the prior work that uses peer-to-peer architecture [14, 3, 13] uses (1), or a variation thereof, to compute trust metrics. However, as the independence assumption does not hold, trust metrics computed in this manner cannot be assigned a precise meaning [12]. Furthermore, the use of (1) is an invitation to several attacks such as free-riding, pseudospoofing, and Sybil attacks. This is to be contrasted with the RS-model of minimum capacity cut measure, where many of these problems are greatly alleviated.

5.2 The Reiter-Stubblebine Model and Metric

Reiter Stubblebine (RS-) metric [12] operates on a directed (acyclic) graph. Before, we go into the details of the model, it is necessary to recall a few concepts from the Graph Theory. Suppose $G(V, E)$ is a directed graph with a source vertex s and a sink vertex t. Suppose each edge e in E has a *capacity* henceforth denoted by $c(e)$.

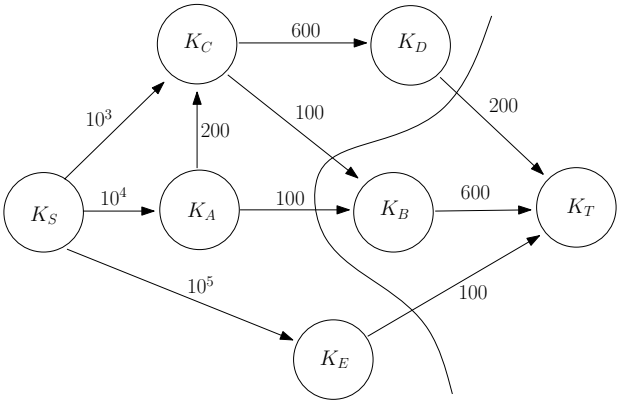

Figure 2: The Reiter-Stubblebine Metric

Definition 1 (Simple-Path Flow) *A* simple-path flow $f(s,t)$ *consists of a simple path*[7] $p(s,t)$ *in* G *that starts at the vertex* s *and ends at the vertex* t *and has a flow value* k *such that* $k \leqslant \min_{e \in p(s,t)} c(e)$.

Definition 2 (Flow) *A* flow $F(s,t)$ *is a set of* n *simple-path flows* $f_1(s,t), f_2(s,t), \ldots, f_n(s,t)$, $n \geqslant 1$, *that satisfy the following constraint imposed by the capacity of the edges in graph* G*:*

$$\sum_{i=1}^{n} I_e(f_i) k_i \leqslant c(e) \qquad \forall e \in E$$

where k_i *is the flow value of* f_i*, and* $I_e(f_i) = 1$ *if the simple path corresponding to* f_i *contains* e*, and* $I_e(f_i) = 0$ *otherwise.*

The flow value of $F(s,t)$ *is given by* $k = \sum_{i=1}^{n} k_i$*. The maximum possible flow value of a flow between* s *and* t *is referred to as the* maximum flow *between* s *and* t *on the graph* G*.*

In Reiter and Stubblebine model, nodes in the graph represent public keys[8], and the edge $e(K_A, K_B)$, denoted by $K_A \rightarrow K_B$ for brevity, exists in the graph if the *user* is in possession of a certificate that assigns attributes (including an owner) to K_B, and whose signature can be verified using K_A.

Assume that the 'capacity' of an edge $K_A \rightarrow K_B$ represents the amount[9] for which the owner of K_A 'insures' the attributes and integrity of K_B. In other words, it is the value for which the owner of K_A will be liable to the user if the attributes bound to K_B in the certificate are incorrect, or if the private key (corresponding to K_B) is used to mislead the user, intentionally or otherwise. It is natural to assume that this numeric value is one of the attributes included in the certificate that the edge represents.

[5] Jøsang [8] proposes using a trust metric that consists of two values representing belief and uncertainty in belief. Several other trust metrics have been proposed, e.g., in [13, 14], they all require discounting and consensus operators to combine two trust metrics into a single one.

[6] In some work [13], operators are defined in such a way that the discounting operator becomes distributive over the consensus operator. This is achieved at the cost of discarding much of the information present in a reputation graph. This, we believe, weakens the natural protection provided by distributed, multiple sources of information in a high threat environment such as the one in which coalition partners are using a MANET.

[7] A simple path has no repeated vertices.

[8] Considering nodes to be public keys explicitly acknowledges the problem of binding a particular key to a particular owner.

[9] The unit of this amount is left unspecified here — it could represent risk tokens as suggested by the Jason report [7], however, it is easier to think in term of hard currency.

The metric is best described using the example in Figure 2. If the attributes bound to K_T (the target public key) turn out to be false, the owners of K_D, K_B, and K_E are each liable to the user for the amount up to 200, 600, and 100 respectively. It is also possible that when the user goes, say, to the owner of K_B, the owner (or its attributes as certified) turns out to be delinquent, and hence the user now is owed by owners of K_A and K_C, for the amount of up to 100 each. Assuming that the owner of K_S is fully trusted, using the above reasoning iteratively, a simple-path flow f from K_S to K_T with a flow value of k can be used to insure an amount of k. This follows by definition since the issuer of a certificate is responsible for an amount up to the 'capacity' indicated in the certificate; and the capacity of each edge on a simple-path flow is at least k providing the user a guarantee of recovery up to the amount k.

By extending this logic further, a flow F between K_S and K_T with flow value k can be used to insure the amount k; and the maximum flow between K_S and K_T is the maximum amount that can be insured using the graph G. By using the Ford-Fulkerson algorithm, the maximum flow can be computed efficiently, and as shown in their famous result, it equals to the value of the *minimum capacity cut* in the graph. For example, in Figure 2, the minimum cut as shown has value 500, and that is the maximum value that can be insured using this graph.

Some of the salient qualities of this metric and model (following the principles enunciated in [12]) are as follows:(a) the user is not required to ascertain name to key bindings to construct the model, except for the root CA, whose name to key binding is reputed, (b) the final metric computed is intuitive, (c) the final metric lets the user ascertain the risk involved in using K_T, (d) the final metric is computed easily using the Ford-Fulkerson algorithm [5, 10], (e) the metric can be computed with partial information, and still give meaningful results, and (f) the insurance metric allows the user to be protected from dependencies in the graph, whether they are unintentional or malicious.

The last point is worth an extra emphasis. Unintentional or malicious dependencies in the reputation graph are at the root of the problems with traditional trust metrics discussed in the earlier section. RS metric has no such dependencies as the issuer of a certificate is ultimately responsible for its use.

6 The New Model and Metric

While the RS-model is attractive, it is not directly applicable to the coalition scenario. Towards that, our proposal is to include two enhancements to the RS-model. First, the RS-model assumes a setting where all transactions between coalition partners can be 'monetized'. As discussed in Section 4, in a joint mission, coalition partners rely on each other for providing heterogeneous, incomparable resources, or services, that cannot be compared with each other. Therefore, instead of assigning a single 'capacity' to each edge, we need to annotate each edge with a vector 'capacity' label so that the i-th coordinate of the capacity vector represents the amount of the i-th resource vouched by the party issuing the certificate. Second, it is natural that each agent issuing a certificate to another agent would compute a probability of certificate misuse. If this probability is included in the certificates, then it can be used to estimate risk; for example, it can be used to calculate the probability that for a given mission, the owner of K_S, would have to pitch in the resources or provide the service due to the failure of other agents. In the following, we discuss these two enhancements in more detail, but before that, we need to extend some graph-theoretic results presented in the previous section.

Suppose $G(V, E)$ is a directed graph with a source vertex s and a sink vertex t. Suppose every edge e in E has an m-dimensional capacity label $\big(c^1(e), c^2(e), \ldots, c^m(e)\big)$ with $c^i(e)$ being the capacity of the edge e for resource R_i. We will refer to a graph of this type as a *multi-resource* graph.

Definition 3 (Multi-Resrouce Simple-Path Flow) *A* simple-path flow $f(s,t)$ *consists of a simple path* $p(s,t)$ *in* G *that starts at the vertex* s *and ends at the vertex* t*, and has a flow value* $k = (k^1, k^2, \ldots, k^m)$ *such that* $k^1 \leqslant \min_{e \in p(s,t)} c^1(e), k^2 \leqslant \min_{e \in ps(s,t)} c^2(e), \ldots, k^m \leqslant \min_{e \in ps(s,t)} c^m(e)$.

Definition 4 (Multi-Resource Flow) *A* flow $F(s,t)$ *is a set of* n *simple-path flows* $f_1(s,t), f_2(s,t), \ldots, f_n(s,t)$, $n \geqslant 1$ *that satisfy the following constraint imposed by the capacity of edges in the graph* G*:*

$$\sum_{i=1}^{n} I_e(f_i) k_i^j \leqslant c^j(e) \qquad \forall e \in E, 1 \leqslant j \leqslant m$$

where k_i *is the flow value of* $f_i(s,t)$*, and* $I_e(f_i) = 1$ *if the simple path corresponding to* f_i *contains* e*, and* $I_e(f_i) = 0$*, otherwise.*

The flow value of $F(s,t)$ *is given by* $k = \sum_{i=1}^{n} k_i = (\sum_{i=1}^{n} k_i^1, \sum_{i=1}^{n} k_i^2, \ldots, \sum_{i=1}^{n} k_i^m)$.

Since the value of a flow in this case is multi-dimensional, it is not straightforward to define the notion of maximum flow on a graph as the flow values cannot be totally ordered. For two flows F_1 and F_2 with flow values k_1 and k_2, we say F_1 has more flow than F_2, $F_1 \geqslant F_2$, if $(k_1^1 \geqslant k_1^2, k_2^1 \geqslant k_2^2, \ldots, k_1^m \geqslant k_2^m)$. Keeping with the intuition of the previous section and generalizing the notion of maximum flow, we want to find a set of flows $\{F_1, F_2, \ldots, F_l\}$ such that $F_1, F_2, \ldots, F_l$ cannot be compared to each other, and for any other flow F_r, there is a flow F_j, for some $j \in \{1, 2, \ldots, l\}$, that has more flow than F_r. It turns out that for any graph, this can be accomplished by a single flow, that is $l = 1$ and there exists a single flow F_{max} such that for any other flow F_r on the graph, $F_{max} \geqslant F_r$.

Theorem 1 *Given a multi-resource graph* G *with a source* s *and a sink* t*, there exists a single flow* F_{max} *such that for any other flow* F_r*,*$F_{max} \geqslant F_r$.

Proof: Consider a flow F^j that maximizes the flow of resource R_j and has zero flow for all other resources. This flow can be computed by considering only the j-th component of the capacity labels and using the Ford-Fulkerson algorithm on the resulting graph. Denote the set of simple-path flows that constitute F^j by Λ^j. Let $k^j(f)$ denote the j-th component of the flow value of a simple-path flow f, and by slight abuse of notation let $k^j(F)$ denote the the j-th component of the flow value of a flow F.

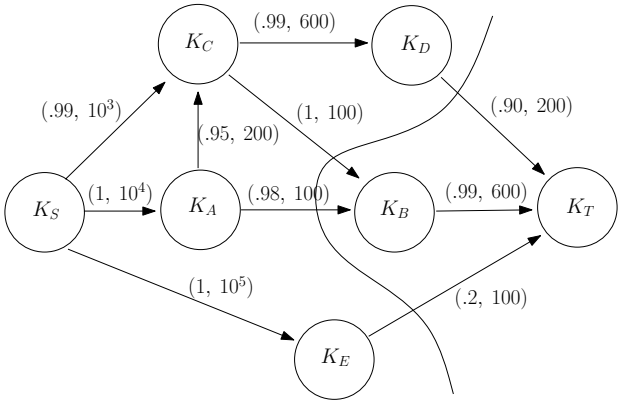

Figure 3: The New Metric

Now consider the set of simple-path flows $\Lambda = \cup_{j=1}^{m} \Lambda^j$. Λ is a flow since it satisfies the required capacity constraint:

$$\sum_{f \in \Lambda} I_e(f) k^j(f) = \sum_{f \in \Lambda^j} I_e(f) k^j(f) \quad (3)$$

$$= k^j(F) \quad (4)$$

$$\leqslant c^j(e) \qquad \forall e \in E, 1 \leqslant j \leqslant m \quad (5)$$

where (3) follows since only the simple-path flows in Λ^j have non-zero flow of R_j, and (4) and (5) follow from the definition of a multi-resource flow as given above.

Clearly, any other flow has less flow value than Λ since its flow value is $(k^1(F^1), k^2(F^2), \ldots, k^m(F^m))$ which, by construction, is maximum possible value in each dimension for the given graph. □

With these definitions and result in place, we are ready to describe the proposed enhancement to the RS model.

6.1 Protocol

On the surface, the protocol is relatively simple—a coalition partner A who wants to vouch for the capacity of another coalition partner B to provide resources $R_1, R_2, \ldots, R_m$ issues a certificate that includes the maximum amount of vouched resources $c(K_A, K_B) = (c^1(K_A, K_B), c^1(K_A, K_B), \ldots, c^m(K_A, K_B))$. As usual, we will omit arguments, K_A, K_B, when they are clear from the context. These certificates may be chained leading to a directed acyclic[10] graph: the nodes in this graph are public keys of the participating entities, and an edge $K_A \rightarrow K_B$ exists in the graph if the user is in possession of a certificate that assigns resource-capacity attributes (and exclusive owner(s)) to K_B, and whose signature can be verified using K_A. Inclusion of the capacity attributes for various resources in a certificate $K_A \rightarrow K_B$ amounts to a guarantee by A that should the capacity attributes bounds in the certificate are found to be incorrect, or if the private key (corresponding to K_B) is used to mislead the user, intentionally or otherwise, then A will instead provide the resource R_i to the user up to the amount c^i.

The certificate corresponding to an edge $K_A \rightarrow K_B$ may include other qualifying attributes. A simple example will be time duration for which the certificate is valid. Here we will focus on the probability $p(K_A, K_B)$ that the issuer A of a certificate $K_A \rightarrow K_B$ computes that B, the owner of K_B, will be able to fulfil its obligations as specified by the attributes in the certificate. Essentially, p represents A's *trust* in certain capabilities of B. How A computes or estimates this trust is left unspecified; it is assumed that each coalition partner will have its own methodology to assess trust. This is one of the key features of the proposed model that it does not require coalition partners to use compatible trust management systems. We will come back to this issue and elaborate it further.

Assume that a coalition partner X wishes to conduct a mission M in collaboration with a coalition partner T, and it relies on T to provide M^i units of resource R_i. Reliance on T to provide resources represents an inherent risk from the perspective of X since T may fail. In order to estimate this risk, either X can rely on its direct past experience with T, or acquire trust in T otherwise. The proposed model of certificate chains is one way to acquire or augment trust for coalition partners.

For instance, Figure 3 shows a graph that includes a probability (trust value of certified party according to certifying party) and a capacity on each edge. For simplicity, we will only consider one resource R_1 at this time. Certificates in the certificate chain $K_S \rightarrow K_A \rightarrow K_B \rightarrow K_T$ vouch for the capability of the certified parties to provide 10^4, 100, 600 units of resource R_1, respectively. Therefore if a coalition partner X has absolute trust in S to honor its obligations, it can use this certificate chain to obtain 100 units of resource R_1. Should T fail to provide the resources, X would be able to rely on certificate $K_B \rightarrow K_T$ and ask B to fulfil the demand for 100 units of resources R_1. According to B, the probability of this scenario unfolding is $(1 - 0.99) = 0.01$. In case, B also fails (according to A, the probability of B failing is 0.02), X has to go to A; and in case, A fails, X would have to go to S and ask for resources. Specifics of risk calculation will depend on the specifics of the scenario—a risk analysis will take into account various probability of failures; it may also take into account additional burden incurred by X in contacting B, A, and S, should it become necessary.

The situation gets more interesting when we consider multiple simple paths from S to the target entity T. As an example, again consider Figure 3. Any simple path P_i from S to T can be treated as a certificate chain, and the minimum value of c on P_i, denoted c_i, is the limit on the amount of R_1 in a mission planning that uses the path P_i. For instance, as computed above, using the middle path K_S, K_A, K_B, K_T, the maximum value is 100, while using the top path K_S, K_C, K_D, K_T, the maximum value is 200. Now, if a user is interested in only 100 units of R_1 from T, then it could go ahead based on any of these paths. On the other hand, if the user is interested in 300 units of R_1, the first 200 can be filled using the top path and the remaining 100 filled using the middle path (or some other path).

In general, the total amount of R_1 that can be reliably obtained from T (assuming S is completely trusted to fulfil its obligations) equals the *maximum flow* in the graph (using the c labels). Generalizing further, by Theorem 1, if S can be fully trusted to fulfil its obligations, then T can be relied to provide resources up to the flow value of maximum flow F_{max} computed on graph G. By Theorem 1, the maximum value for resource R_i can be computed

[10] Without any loss of generality we can assume that the graph is acyclic.

by considering only the i-th component of the capacity labels. If each certificate also includes trust values, then X can also conduct a more thorough risk analysis on relying on T to provide resources. The risk computed will depend on X's own assumptions of trustworthiness of other parties, and how its risk framework combines these trust values to compute risk.

We note that the system allows for a natural feedback mechanism. In case of a good 'transaction' between T and X, T would want to provide evidence to the parties that certified it which would then propagate this evidence up the certificate chain. On the other hand, in case of a bad transaction, X has incentive to notify parties up in the certificate chain. In this way, for both good or bad transaction, a certificate issuing party will receive a feedback that can be used to refine trust values in the later rounds. Note that the auditing and reporting mechanism may not work on a transaction by transaction basis, but may rely on batch processing, in which case, there will be a delay incurred in the feedback mechanism. However, such delays are manageable in a trust management systems as shown by Srivatsa *et al.* [2].

7 Discussion

Let us examine some potential implications of our proposal. Our model requires that if there is an edge in the reputation graph from K_A to K_B labeled with capacity c for resource R, then the owner of the public key K_A is responsible for provide up to c units of the resource R if B fails. This liability raises the question of motivation on behalf of certifying parties. We note that this question is not unique to our proposal, it is a question for all trust management systems. In traditional trust management systems [13, 14] vouching false or misleading attributes to other parties incurs no direct penalty, and as a consequence, these systems suffer from a number of attacks. Our system avoids this problem by holding the certifying party directly responsible for its certificates.

Our proposal naturally provides separation of a multitude of concerns. In our proposal, a coalition partner that vouches for another are not assumed to be next neighbors on a MANET. In contrast to traditional proposals [3, 13], our proposal does not require that agents who are competent in MANET communication also be competent in providing recommendations. The proposal separates communication protocol issues from trust related issues.

Another benefit of our proposal is that the input provided by a malicious node to the trust management system does not get intermixed with input provided by other nodes. In fact, all trust values and capabilities are included in certificates that remain separate from each other. Only when needed these certificates are exchanged between partners to allow risk analysis. If due to privacy or secrecy reasons, a coalition partner does not want to reveal its certificate chains, then it can enter in a trust negotiation with another partner and provide these certificate chains piecemeal until the risk is sufficiently mitigated to conduct a mission.

In our proposal, agents need not be homogenous, they need not to use the same or similar criterion to evaluate trust in others. The proposed model is a hybrid where trust is manifested in two forms: a more traditional non-binding trust that is expressed as a single value between zero and one, and a trust that requires trusting agent to take explicit responsibility of the trustee's actions. We believe this provides a better compromise and accommodation of different types and degrees of trust.

8 Acknowledgment

This work immensely benefited from many discussions held in various meetings and workshops organized by "International Technology Alliance". In particular, the authors would like to acknowledge numerous discussions held with Shane Balfe, Greg Crinicione, Virgill Gligor, Jorge Lobo, Kenny Paterson, and Brian Reviara.

References

[1] IS-100: Introduction to incident command system, 2005. Available at www.training.fema.gov.

[2] A trust management system for securing information flows. In *16th ACM Conference on Computer and Communication Security (CCS)*, 2008.

[3] W. J. Adams. *Decentralized trust-based access control for dynamic collaborative environments*. PhD thesis, Virginia Polytechnic Institute and State University, Mar. 2006.

[4] J. A. Clark, H. R. Chivers, J. Murdoch, and J. A. McDermid. The collaboration lifecycle. ITA TA2 Workshop, CUNY, NY, USA, 2006.

[5] L. R. Ford Jr. and D. R. Fulkerson. Maximum flow through a network. *Canadian Journal of Mathematics*, 8:399–404, 1956.

[6] Y.-T. C. Hung, A. R. Dennis, and L. Robert. Trust in virtual teams: Towards an integrative model of trust formation. In *HICSS*, 2004.

[7] Jason Program Office, MITRE Corporation. *Horizontal Integration: Broader access models for realizing information dominance*, 2004.

[8] A. Jøsang. A logic for uncertain probabilities. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 9(3):279–212, 2001.

[9] D. H. McKnight and N. L. Chervany. The meanings of trust. Technical Report Technical Report 94-04, Carlson School of Management, University of Minnesota, Jan. 1996.

[10] C. Papadimitriou and K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity*. Printice Hall, 1982.

[11] R. E. Petty and J. T. Cacioppo. *Communication and Persuasion: Central and Peripheral Routes to Attitude Change*. Springer-Verlag, New York, 1986.

[12] M. K. Reiter and S. G. Stubblebine. Toward acceptable metrics of authentication. In *IEEE Symp. on S&P*, pages 10–20, 1997.

[13] G. Theodorakopoulos and J. S. Baras. On trust models and trust evaluation metrics for ad hoc networks. *IEEE Journal on Selected Areas in Communications*, 24(2):318–328, February 2006.

[14] L. Xiong and L. Liu. PeerTrust: Supporting reputation-based trust for peer-to-peer electronic communities. *IEEE Trans. Knowl. Data Eng.*, 16(7):843–857, 2004.